

MOTION VECTOR CODING METHOD

BACKGROUND OF THE INVENTION

This application claims the benefit of priority based on the U.S.

Provisional Application No. 60/206,783 filed on May 24, 2000 and claims the benefit of foreign priority under 35 U.S.C. § 119 from Korean Patent

5 Application No. 2000-58757 filed on October 6, 2000. Both Provisional Application No. 60/206,783 and Korean Patent Application No. 2000-58757 are incorporated herein by reference.

1. Field of the Invention

The present invention relates to a motion vector coding method and an
10 affine motion estimation method, and more particularly, to a motion vector coding method capable of producing a low bit rate bitstream and an affine motion estimation method capable of effectively performing coding on a small block.

2. Description of the Related Art

15 Recently, the International Telecommunication Union-Terminal Sector (ITU-T) has made efforts to standardize the H.26L protocol for next generation visual telephony. Since standard schemes such as H.261, H.263, and H.263+ that define a motion vector coding method for visual telephony were adopted as an ITU-T standard, technologies based on H. 263++ and
20 Moving Picture Experts Group (MPEG)-4 standard schemes have been

developed. Nonetheless, there remains a need for further improving coding efficiency for ultra low speed real-time applications and a short end-to-end delay. That is, it is highly desirable to have an estimation method and a motion vector coding method that provide for an improved frame rate at the same bit rate as that of the coding method of the H.263+ standard, or that significantly reduces a bit rate while maintaining the same image quality as that encoded by the H.263+ compliant coding method.

In one conventional motion estimation method, assuming that the pair (i,j) are the coordinates of a macroblock or a sub-block thereof, an affine motion estimation is performed to represent the motion of a pixel in an image using the following Equations (1a) and (1b):

$$v_x(i,j) = a_0 + a_1 i + a_2 j \quad \dots(1a)$$

$$v_y(i,j) = a_3 + a_4 i + a_5 j \quad \dots(1b)$$

where $v_x(i,j)$ and $v_y(i,j)$ are motion magnitude components of the X- and Y-axis directions for a pixel located in the block (i,j) . The expression $(v_x(i,j), v_y(i,j))^T$, consisting of motion magnitude components $v_x(i,j)$ and $v_y(i,j)$, is referred to as the motion vector of the pixel located at the coordinates (i,j) . That is, the motion vector of each pixel is determined by each pixel location and six parameters $(a_0, a_1, a_2, \dots, a_5)^T$. These parameters $(a_0, a_1, a_2, \dots, a_5)^T$ may be called affine motion parameters. However, according to a method of estimating motion using the affine motion parameters, as the number of bits representing the affine motion parameters increases, computation for a motion

estimation becomes more complex and takes more time. Furthermore, with respect to some blocks, this affine motion estimation cannot be more effective than conventional translational motion estimation.

Meanwhile, standards such as H.261, H.263, MPEG-1, and MPEG-2 represent the motion of a pixel based on a translational motion model expressed by:

$$v_x(i,j) = t_1 \quad \dots(2a)$$

$$v_y(i,j) = t_2 \quad \dots(2b)$$

As is evident from Equations (2a) and (2b), the motion vectors of all pixels in a block are fixed as one vector. However, in the case of affine motion, as expressed in Equations (1a) and (1b), a motion vector with respect to each pixel location is variable. The affine motion estimation is capable of representing complex motions that include any or all of translations such as rotation, magnification, reduction and shear, thereby allowing for more precise motion estimation.

To estimate the motion of an image using the affine motion estimation, the affine motion parameters as expressed in Equations (1a) and (1b) must be obtained on a block-by-block basis. The motion parameters correspond to displacements that minimize the difference between pixel values for a present image and a preceding motion-compensated image and are expressed by the following Equation (3):

$$\arg \min_{a_k} \sum_{(i,j) \in M_k} \{I_n(i,j) - I_{n-1}(i + v_x(i,j) + v_y(i,j))\}^2 \quad \dots(3)$$

where $I_n(i,j)$ denotes the luminance of the pixel at location (i,j) , and M_k denotes the k^{th} block. $(v_x(i,j), v_y(i,j))^T$ is expressed by Equations (1a) and (1b). That is, motion parameters that minimize the luminance difference between a present block and a previous block motion-compensated by the motion parameters are expressed in Equation (3), and the motion parameters are called motion-estimated parameters.

In the affine motion estimation method according to the conventional art, the motion parameters are obtained using the following Equation (4):

$$a_k^{l+1} = a_k^l + u_k^l \quad \dots(4)$$

where the term $a_k^l = (a_0, a_1, a_2, a_3, a_4, a_5)^T$, and l denotes an iteration coefficient.

When l equals 0, the motion parameter is expressed by:

$$a_k^0 = (0,0,0,0,0,0)^T$$

In this case, the motion parameter is called an initial value.

$$u_k^1 = \left\{ \sum_{(i,j) \in Mk} \{h_{ij}^1 (h_{ij}^1)^T\} \right\}^{-1} \left\{ \sum \{d_n^1(i,j)^T h_{ij}^1\} \right\}$$

$$h_k^1 = (G_x^1(i,j), iG_x^1(i,j), jG_x^1(i,j), G_y^1(i,j), iG_y^1(i,j), jG_y^1(i,j))^T$$

$$d_n^l(i, j) = I_n(i, j) - I_{n-1}(i, j)(i + a_0^l + a_1^l i + a_2^l j + a_3^l + a_4^l i + a_5^l j) \quad \dots(8)$$

$$G_X^l(i, j) = \{I_n(i + 1, j) - I_{n-1}(i, j)(i + a_0^l + a_1^l i + a_2^l j - 1, j + a_3^l + a_4^l i + a_5^l j)\} / 2 \quad \dots(9a)$$

$$G_Y^l(i, j) = \{I_n(i, j + 1) - I_{n-1}(i, j)(i + a_0^l + a_1^l i + a_2^l j - 1, j + a_3^l + a_4^l + a_5^l j - 1)\} / 2 \quad \dots(9b)$$

The method of estimating the affine motion parameters shown in Equations 4 through 9a and 9b is called a differential motion estimation method. The differential motion estimation method is mostly used in affine motion estimation.

5 In this case, for affine motion estimation, first, the iteration coefficient l is set to “0” and the square error is set at a maximum as possible. Next, the value obtained from Equation (6) is updated using Equations 6 through 9a and 9b, and then the updated value is substituted into Equation (4) to thereby obtain a_k^{l+1} . Then, the difference between the present block and the motion-
10 compensated previous block is obtained using the value a_k^{l+1} and the following Equation:

$$SE(l + 1) = \sum_{i,j \in M_k} \{I_n(i, j) - I_{n-1}(i + a_0^{l+1} + a_1^{l+1} i + a_2^{l+1} j, j + a_3^{l+1} + a_4^{l+1} i + a_5^{l+1} j)\} \quad \dots(10)$$

If $SE(l+1)$ is less than $SE(l)$, l increases by 1 and the above steps are repeatedly performed. If $SE(l+1)$ is greater than $SE(l)$, a_k^l at that time is determined as an estimated motion parameter, and then repetitions of the
15 motion estimation process are terminated.

However, the affine motion estimation method has a problem in that motion estimation performance for small-sized blocks is significantly degraded. Thus, although the affine motion estimation method exhibits an excellent image prediction over the translational motion estimation method, it cannot be applied
5 to standards such as H.261, H.263, MPEG-1, and MPEG-4.

Meanwhile, the affine motion parameters are real numbers with decimal points. Thus, to use these parameters in actual video coding, they must be converted or quantized to fixed-point numbers.

In standards such as H.261, H.263, MPEG-1, and MPEG-4, to which a
10 translational motion model is applied, motion information to be encoded in each block is predictively encoded using motion vectors, thereby reducing the number of bits generated per unit time in a bitstream during coding of the motion information. However, if an affine motion model is used, since each of the six motion parameters shown in Equations (1a) and (1b) is not affected by
15 neighboring blocks, it is very difficult to predictively encode motion information as employed in the translational motion model. That is, a significantly large number of bits may be required in coding a motion parameter of an affine motion model. Thus, it is highly desirable to have a method of effectively coding affine motion parameters.

20

SUMMARY OF THE INVENTION

To solve the above problems, it is an objective of the present invention to provide a method of coding a motion vector that reduces a generated bit rate

while maintaining motion estimation performance available from an affine motion estimation.

It is another objective of the present invention to provide a method of estimating motion that is capable of effectively performing coding on a small
5 block.

It is still another objective of the present invention to provide a motion vector decoding method for decoding a bitstream encoded by the motion vector coding method.

Accordingly, to achieve the above objectives, the present invention
10 provides a method of coding a motion vector including the steps of (a) performing an affine motion estimation to obtain affine motion parameters, (b) converting the affine motion parameters to a predetermined number of translational motion vectors, and (c) coding the difference between the converted translational motion vectors of a current block and the converted
15 translational motion vectors of a previous block.

The motion vector coding method further includes the step of quantizing the translational motion vectors in the step (b) to fixed-point numbers having a predetermined accuracy prior to the step (c). Furthermore, the motion vector coding method further includes the step of determining a
20 predetermined pixel range for each translational motion vector of the current block and obtaining an accuracy ϕ for each pixel value in the predetermined range, the accuracy ϕ producing the smallest value in $\min_{\phi \in \Pi} \{MSE(\phi) + \lambda \text{Bits}(\phi)\}$

among a set Π of accuracies, where QP is a quantization interval used in coding an image, λ is a constant determined based on the quantization interval QP , $MSE(\varphi)$ denotes the mean square sum of a differential signal between the current block and the preceding motion-compensated block when translational motion vectors of the current block are represented with an accuracy φ , and Bits(φ) denotes the number of bits used to code the translational motion vector of the current block with an accuracy φ , prior to the step (c).

The method further includes the step of predicting the converted translational motion vectors of the current block by translational motion vectors of neighboring blocks using $\hat{V}_{x,Ca} = \text{median}\{V_{x,Cb}, V_{x,Uc}, V_{x,Ud}\}$, $\hat{V}_{y,Ca} = \text{median}\{V_{y,Cb}, V_{y,Uc}, V_{y,Ud}\}$, $\hat{V}_{x,Cb} = \text{median}\{V_{x,Ca}, V_{x,Ud}, V_{x,Dc}\}$, $\hat{V}_{y,Cb} = \text{median}\{V_{y,Ca}, V_{y,Ud}, V_{y,Dc}\}$, $\hat{V}_{x,Cc} = \text{median}\{V_{x,Ld}, V_{x,Ca}, V_{x,Cb}\}$, and $\hat{V}_{y,Cc} = \text{median}\{V_{y,Ld}, V_{y,Ca}, V_{y,Cb}\}$, where a hat denotes an operation for obtaining a predicted value and median { } denotes an operation of obtaining the translational motion vector having the median magnitude among the translational motion vectors.

Step (c) includes the steps of: (c-1) calculating the differences between the translational motion vectors of the current block and the blocks neighboring the current block by using the converted translational motion vectors of the current block and the translational motion vectors of the neighboring blocks;

and (c-2) performing variable length coding on the obtained differences between the translational motion vectors.

Step (a) includes the step of obtaining six affine motion estimation parameters $(a_0, a_1, a_2, \dots, a_5)^T$ by estimating the motion of a pixel in an image using pixel values of a block to be encoded in a current frame and pixel values in a preceding frame and using, $v_x(i, j) = a_0 + a_1 i + a_2 j$, and $v_y(i, j) = a_3 + a_4 i + a_5 j$, where the pair (i,j) are the coordinates of a macroblock or a sub-block thereof.

Step (b) includes the step of obtaining motion vectors of the center points of the sub-blocks A, B, C, and D by using $(v_{x,A}, v_{y,A}) = (a_0 + a_1 \alpha + a_2 \alpha, a_3 + a_4 \alpha + a_5 \alpha)$, $(v_{x,B}, v_{y,B}) = (a_0 + 3a_1 \alpha + a_2 \alpha, a_3 + 3a_4 \alpha + a_5 \alpha)$, and $(v_{x,C}, v_{y,C}) = (a_0 + a_1 \alpha + 3a_2 \alpha, a_3 + a_4 \alpha + 3a_5 \alpha)$ based on

$$v_x(i, j) = \frac{1}{2\alpha}(4\alpha - i - j)v_{x,A} + \frac{1}{2\alpha}(-2\alpha + i)v_{x,B} + \frac{1}{2\alpha}(-2\alpha + j)v_{x,C},$$

$$v_y(i, j) = \frac{1}{2\alpha}(4\alpha - i - j)v_{y,A} + \frac{1}{2\alpha}(-2\alpha + i)v_{y,B} + \frac{1}{2\alpha}(-2\alpha + j)v_{y,C}.$$

where one macroblock consists of sub-blocks A, B, C, and D, the size of the macroblock is $S \times S$, and the constant α is $S/4 + 1/2$.

After step (b), the motion vector coding method further includes the steps of (b'-1) quantizing the converted translational motion vectors to fixed-point numbers having a predetermined accuracy, (b'-2) selecting an optimum accuracy considering a bit rate and distortion among accuracies, with which the

translational motion vectors are represented, and (b'-3) predicting translational motion vectors having the selected accuracy using converted translational motion vectors of neighboring blocks.

Prior to step (a), the method further includes the step of obtaining one
 5 translational motion vector for the current block by a block matching technique, and step (a) includes the step (a') performing an affine motion estimation using the obtained translational motion vector as an initial value.

Alternatively, prior to step (a), the method further includes the step of obtaining a motion parameter that minimizes the mean square sum of a
 10 difference signal between the current block and the motion-compensated previous block among estimated affine motion parameters of blocks located above the current block and located on the left thereof, and step (a) includes the step of (a') performing an affine motion estimation using the obtained motion parameter as an initial value to obtain affine motion parameters for a
 15 predetermined pixel range.

The present invention also provides a method of coding a motion vector including the steps of (a) performing an affine motion estimation to obtain affine motion parameters, (b) obtaining translational motion vectors based on

$$v_x(i, j) = \frac{1}{2\alpha} (4\alpha - i - j) v_{x,A} + \frac{1}{2\alpha} (-2\alpha + i) v_{x,B} + \frac{1}{2\alpha} (-2\alpha + j) v_{x,C} \quad \text{and}$$

$$v_y(i, j) = \frac{1}{2\alpha} (4\alpha - i - j) v_{y,A} + \frac{1}{2\alpha} (-2\alpha + i) v_{y,B} + \frac{1}{2\alpha} (-2\alpha + j) v_{y,C}.$$

size of a block is $S \times S$, and the constant α is $S/4 + 1/2$, and (c) coding the difference between the translational motion vectors obtained in step (b).

The present invention also provides a method of coding a motion vector including the steps of: (a) determining a predetermined pixel range for
5 each translational motion vector of a current block; (b) determining the translational motion vector of a current block for each pixel value in the predetermined range, wherein the pixel motion vector is represented with an accuracy φ that produces the smallest value in $\min_{\varphi \in \Pi} \{MSE(\varphi) + \lambda \text{Bits}(\varphi)\}$ among a set Π of accuracies, where QP is a quantization interval used in
10 coding an image, λ is a constant determined based on the quantization interval QP , $MSE(\varphi)$ denotes the mean square sum of a differential signal between the current block and the preceding motion-compensated block when translational motion vectors of the current block are represented with an accuracy φ , and $\text{Bits}(\varphi)$ denotes the number of bits used to code the translational motion
15 vector of the current block with an accuracy φ ; and (c) coding the difference between the translational motion vectors of the current block and the translational motion vectors of the previous block.

The present invention also provides a method of coding a motion vector including the steps of: (a) determining a predetermined pixel range for each of
20 six translational motion vector components $\{v_{X,A}, v_{Y,A}, v_{X,B}, v_{Y,B}, v_{X,C}, v_{Y,C}\}$; (b) obtaining the translational motion vector of a current block for each of the pixel values in the predetermined range determined for each of the six

translational motion vector components $\{v_{X,A}, v_{Y,A}, v_{X,B}, v_{Y,B}, v_{X,C}, v_{Y,C}\}$ in step (a), wherein the pixel motion vector is represented with an accuracy φ that produces the smallest value in $\min_{\varphi \in \Pi} \{MSE(\varphi) + \lambda \text{Bits}(\varphi)\}$ among a set Π of accuracies, where QP is a quantization interval used in coding an image, λ is a constant determined based on the quantization interval QP, MSE(φ) denotes the mean square sum of a differential signal between the current block and the preceding motion-compensated block when translational motion vectors of the current block are represented with an accuracy φ , and Bits (φ) denotes the number of bits used to code the translational motion vector of the current block with an accuracy φ ; and (c) coding the difference between the translational motion vectors obtained in step (b).

The present invention also provides a motion estimation method including the steps of (a) performing an affine motion estimation to obtain affine motion parameters, and (b) converting the affine motion parameters to a predetermined number of translational motion vectors.

The present invention also provides a method of decoding a motion vector including the steps of (a) receiving encoded data, (b) decoding the received data to obtain translational motion vectors, (c) converting the obtained translational motion vectors to affine motion parameters, and (d) performing motion compensation using the obtained affine motion parameters.

BRIEF DESCRIPTION OF THE DRAWINGS

The above objectives and advantages of the present invention will become more apparent by describing in detail a preferred embodiment thereof with reference to the attached drawings in which:

5 FIG. 1 is a flowchart showing the main steps of a motion vector coding method according to an embodiment of the present invention;

 FIG. 2 shows the coordinate system of an affine motion vector used in a motion estimation method according to an embodiment of the present invention;

10 FIG. 3 shows the codes of blocks; and

 FIG. 4 is a flowchart showing the main steps of a motion vector coding method according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Referring to FIG. 1, according to a method of coding a motion vector
15 according to the present invention, first, pixel values of a block to be coded in a current image frame and pixel values of a search area in a preceding image frame are used to perform an affine motion estimation, thus obtaining affine motion parameters of the current block (step 102). The affine motion estimation may be described using six parameters, for example, a_0 , a_1 , a_2 , a_3 , a_4 ,
20 and a_5 . In this embodiment, the affine motion is described using three translational motion vectors. For this purpose, next, the affine motion parameters obtained by performing the motion estimation are converted to

translational motion vectors (step 104). In step 104, assuming that the size of the current block and a constant α are defined as $S \times S$, and $S/4 + 1/2$, respectively, the affine motion parameters are converted to translational motion vectors based on the following Equations (11a) and (11b):

$$v_x(i,j) = \frac{1}{2\alpha}(4\alpha - i - j)v_{x,A} + \frac{1}{2\alpha}(-2\alpha + i)v_{x,B} + \frac{1}{2\alpha}(-2\alpha + j)v_{x,C} \quad \dots(11a)$$

$$v_y(i,j) = \frac{1}{2\alpha}(4\alpha - i - j)v_{y,A} + \frac{1}{2\alpha}(-2\alpha + i)v_{y,B} + \frac{1}{2\alpha}(-2\alpha + j)v_{y,C} \quad \dots(11b)$$

5

Since the affine motion parameters are real numbers with decimal points, they are converted or quantized to fixed-point numbers so as to be used in actual motion vector coding. For this purpose, the translational motion vectors converted in step 104 are quantized to fixed-point values having a
10 predetermined accuracy (step 106).

Assuming that the current block consists of four sub-blocks A, B, C, and D as shown in FIG. 2, if the coordinate values of the center points of the sub-blocks A, B, C and D are substituted into the coordinates i and j of Equations (1a) and (1b), the motion vectors of the center points of the sub-
15 blocks A, B, C, and D can be obtained as follows:

$$[v_{x,A}, v_{y,A}] = [a_0 + a_1\alpha + a_2\alpha, a_3 + a_4\alpha + a_5\alpha] \quad \dots(12a)$$

$$[v_{x,B}, v_{y,B}] = [a_0 + 3a_1\alpha + a_2\alpha, a_3 + 3a_4\alpha + a_5\alpha] \quad \dots(12b)$$

$$[v_{x,C}, v_{y,C}] = [a_0 + a_1\alpha + 3a_2\alpha, a_3 + a_4\alpha + 3a_5\alpha] \quad \dots(12c)$$

Next, an optimum accuracy is selected (step 108) based on:

$$\min_{\varphi \in \Pi} \{MSE(\varphi) + \lambda Bits(\varphi)\} \quad \dots(13)$$

where φ and Π are a selected accuracy and a set of accuracies, respectively.

5 Furthermore, λ is set to $0.85QP^2$ assuming that QP is a quantization interval used in coding an image. $MSE(\varphi)$ denotes the mean square sum of a differential signal between the current block and a preceding motion-compensated block when translational motion vectors of the current block are represented with the accuracy φ . $Bits(\varphi)$ denotes the number of bits used to
10 code the translational motion vectors of the current block with the accuracy φ . Thus, the optimum accuracy means accuracy φ , which produces the smallest value in Equation (13) among the set Π of accuracies. The translational motion vectors of the current block are represented with the optimum accuracy.

Next, the converted translational motion vectors of the current block are
15 predicted using translational motion vectors of neighboring blocks, by the following Equations (step 110):

$$\hat{v}_{X, Ca} = \text{median}\{v_{X, Cb}, v_{X, Ue}, v_{X, Ud}\} \quad \dots(14a)$$

$$\hat{v}_{Y, Ca} = \text{median}\{v_{Y, Cb}, v_{Y, Ue}, v_{Y, Ud}\} \quad \dots(14b)$$

$$\hat{v}_{X, Cb} = \text{median}\{v_{X, Ca}, v_{X, Ud}, v_{X, De}\} \quad \dots(15a)$$

$$\hat{v}_{Y,Cb} = \text{median}\{v_{Y,Ca}, v_{Y,Ud}, v_{Y,Dc}\} \quad \dots(15b)$$

$$\hat{v}_{X,Cc} = \text{median}\{v_{X,Ld}, v_{X,Ca}, v_{X,Cb}\} \quad \dots(16a)$$

$$\hat{v}_{Y,Cc} = \text{median}\{v_{Y,Ld}, v_{Y,Ca}, v_{Y,Cb}\} \quad \dots(16b)$$

In association therewith, FIG. 3 shows the codes of the blocks. In Equations (16a) and (16b), a hat denotes an operation for obtaining a predicted value, and median { } denotes an operation of obtaining a translational motion
5 vector having a median magnitude among three translational motion vectors. Since the affine motion parameters are represented using the three translational motion vectors as expressed in Equations (11a) and (11b), the translational motion vector of a sub-block at the lower right of the current block, for example, a sub-block “d” may not be predicted nor encoded. Thus, sub-blocks
10 of all blocks at the lower right may not be encoded. However, as expressed in Equations 14a through 16b, the translational motion vectors of sub-blocks Ud and Ld are used in predicting the converted translational motion vectors of the current block. The translational motion vector of the sub-block “d” is calculated using the following Equations (17a) and (17b):

$$V_{X,Ud} = V_{X,Ub} + V_{X,Uc} - V_{X,Ua} \quad \dots(17a)$$

$$V_{Y,Ud} = V_{Y,Ub} + V_{Y,Uc} - V_{Y,Ua} \quad \dots(17b)$$

Furthermore, if an accuracy for the translational motion vectors of the neighboring blocks in the Equations 14a through 16b is different from the accuracy for the translational motion vectors of the current block, the former is preferably revised to be the same as the latter.

Next, the difference between translational motion vectors is calculated using the converted translational motion vectors of the current block and the translational motion vectors of the neighboring blocks (step 112), and variable length coding is performed on the calculated translational motion vector difference (step 114).

Table 1

Test image sequence	Format	Translational motion vector	Affine motion vector
Foreman	QCIF, 10Hz	7.35	9.03
Container	QCIF, 10Hz	2.79	4.72
News	QCIF, 10Hz	4.14	6.30
Silent	QCIF, 15Hz	4.43	7.16
Paris	CIF, 15Hz	4.58	5.42
Mobile	CIF, 30Hz	4.08	6.22
Tempete	CIF, 10Hz	5.25	9.62

Table 1 contains the results of evaluating an average motion vector entropy using test images in a motion vector coding method according to an embodiment of the present invention. As is evident from Table 1, the average motion vector entropy for test image sequences when the motion vector coding method according to the present invention based on an affine motion estimation is applied is greater than that for test image sequences when a conventional motion vector coding method based on a translational motion vector estimation is applied.

Table 2

Test image sequence	Format	16x16 block matching	8x8 block matching	Present invention
Foreman	QCIF, 10Hz	29.34	31.58	32.37
Container	QCIF, 10Hz	38.31	39.01	39.93
News	QCIF, 10Hz	31.52	33.38	33.93
Silent	QCIF, 15Hz	33.53	35.45	35.98
Paris	CIF, 15Hz	31.04	32.49	32.52
Mobile	QCIF, 30Hz	26.85	28.14	28.62
Tempete	QCIF, 10Hz	28.48	29.59	29.69

Furthermore, Table 2 shows the results of measuring a peak-signal-to-noise ratio (PSNR) indicating an average prediction gain, which is expressed in decibels (dB). As is evident from Table 2, the PSNR measured for test image sequences when the motion vector coding method based on the affine motion estimation is applied is greater than that measured for test image

sequences when the conventional motion vector coding method based on the translational motion vector estimation is applied.

That is, if a method including the steps 102 and 104 is applied as a motion estimation method for calculating a translational motion vector, high performance motion estimation is performed on small blocks whose dimensions are less than 16 by 16 (16 x 16) pixel blocks while maintaining improved motion estimation performance available from an affine motion estimation.

Furthermore, the motion vector coding method according to the present invention allows for bit generation ranging from 9.62 x 3 or 28.86 bits at the high end to 4.72 x 3 or 14.16 bits at the low end, thus reducing the bit rate compared to a conventional method in which 6 x 6 or 36 bits are generated during fixed length coding (FLC) of each affine parameter using 8 bits.

In the above embodiment described with reference to FIG. 1, step 102 of performing an affine motion estimation using pixel values of a current block to be coded in a current image frame and of a search region in a preceding image frame to obtain affine motion parameters of the current block, the step 104 of converting the affine motion parameter to translational motion vectors, and the step 106 of quantizing the converted translational motion vectors to fixed-point numbers having a predetermined accuracy are performed in a separate manner.

However, the above two steps 102 and 104 can be integrated into one step by performing an affine parameter estimation based on the Equations (11a) and (11b), which reduces the complexity of the overall process.

Alternatively, the above three steps 102, 104 and 106 may be
5 integrated into one step by determining a predetermined pixel range for each translational motion vector of the current block and then determining the translational motion vector for each pixel value in the predetermined range, the pixel motion vector being represented with an accuracy that produces the smallest value in Equation (13). For example, it is possible to determine a
10 predetermined subpixel range and then to determine a subpixel motion vector of the translational motion vector for each subpixel value in the predetermined range, the subpixel motion vector being obtained with an accuracy that produces the smallest value in Equation (13). In this case, the complexity of the process is further reduced.

15 Furthermore, in step 102 in the embodiment described with reference to FIG. 1, each translational motion vector component is obtained considering all variable factors that the component may have. However, it is possible to determine a predetermined range of pixels for each of six translational motion vector components $\{v_{X,A}, v_{Y,A}, v_{X,B}, v_{Y,B}, v_{X,C}, v_{Y,C}\}$ and to determine a pixel
20 motion vector of the translational motion vector for each pixel value in the predetermined range, the translational motion vector being obtained with an accuracy that gives the smallest value in Equation (13).

Furthermore, in step 102, an initial value is set to “0” as expressed by Equation (5). However, it is preferable to perform step 102 by using one translational motion vector, which is obtained by means of a block matching algorithm for the current block, as the initial value. Furthermore, it is possible
5 to use a motion parameter which minimizes the mean square sum of the difference signal between the current block and the preceding motion-compensated block among the estimated affine motion parameters of blocks located above the current block and located on the left thereof.

Furthermore, it is possible to reduce the complexity by obtaining one
10 translational motion vector for the current block by a block matching algorithm and then using the obtained translational motion vector as the initial value to obtain the affine motion parameters for a predetermined pixel range in step 102.

Furthermore, it is possible to further reduce the complexity by
15 obtaining a motion parameter that minimizes the mean square sum of a difference signal between the current block and the preceding motion-compensated block among the estimated affine motion parameters of blocks located above the current block and located on the left thereof, and then using the obtained motion parameter as an initial value to obtain affine motion
20 parameters for a predetermined pixel range in the step 102.

Meanwhile, it is possible to decode a bitstream encoded using the motion vector coding method according to the present invention by motion vector decoding according to the present invention. FIG. 4 is a flowchart

showing the main steps of motion vector decoding according to an embodiment of the present invention. Referring to FIG. 4, during the motion vector decoding according to the present invention, variable length coded data is received (step 402), and then the received data is variable length decoded to
5 obtain translational motion vectors (step 404). The bit stream encoded by the motion vector coding method contains the difference between median values of translational motion vectors of a corresponding block and its neighboring blocks. Thus, after performing variable length decoding on the received data to obtain the difference between the median values of the corresponding block
10 and the neighboring blocks, the translational motion vectors are obtained using the difference.

Next, the obtained translational motion vectors are converted to affine motion parameters (step 406). The step 406 is a reverse process of the step of converting the motion parameters to the motion vectors described with
15 reference to FIGS. 1 and 2, and thus a detailed explanation is omitted. Then, motion compensation is performed using the affine motion parameters obtained in step 406 to restore an image (step 408).

Furthermore, the motion vector coding method, the motion estimation method, and motion vector decoding according to the present invention can be
20 written as a program executed on a personal or server computer. The program codes and code segments constructing the program can be easily inferred by computer programmers in the industry. Furthermore, the program can be stored in a computer-readable recording medium. The recording medium

includes, among other things, a magnetic recording medium, an optical recording medium, and a radio medium.

As described above, the motion vector coding method according to the present invention allows for a low bit generation while maintaining improved
5 motion estimation performance that can be obtained by using an affine motion estimation. Furthermore, the motion vector coding method according to the present invention is capable of performing high performance motion estimation on small blocks whose dimensions are less than a 16 x 16 pixel block.